Claim Amendments

Applicants have amended claim 1, 5 and 57, and have added new claim 59. Applicants set forth below a complete listing of the claims with the corresponding status indicated for each claim.

1. (Currently Amended) A method for providing a design test bench, the method comprising:

providing a single executable program ~~that partitions functionality of the test bench between a simulation engine and one or more scripted routines, wherein each scripted routine implements a corresponding function;~~

~~instantiating one or more interpreters in the simulation engine, wherein each interpreter is associated with a corresponding scripted routine and may interact with the simulation engine independently of any other interpreter~~ adapted to create a primary thread and a secondary thread, the primary thread running a simulation engine, the secondary thread running an interpreter that interprets a scripted routine comprising a user-defined call that is mapped to a simulation engine task;

providing a user-defined simulation engine function associated with the interpreter;

upon encountering the user-defined function, passing control from ~~causing~~ the simulation engine ~~to pass control~~ to the ~~corresponding~~ interpreter to interpret the scripted routine ~~upon encountering one of the functions~~; and

upon encountering the user-defined call, passing control from the ~~causing the corresponding~~ interpreter ~~to return control~~ to the simulation engine ~~upon encountering a task that is performed by the simulation engine~~.

2-4. (Cancelled).

5. (Currently Amended) The method of Claim 1, further comprising synchronizing the simulation engine and the ~~corresponding~~ interpreter via [[a]] semaphores.

6-56. (Cancelled).

57. (Currently Amended) The method of Claim 1, further comprising directly sharing variables between the simulation engine and the ~~one or more~~ scripted routines.

58. (Cancelled).

59. (New) A method for providing a design test bench, the method comprising:

providing a single executable program adapted to create a primary thread and one or more secondary threads, the primary thread running a simulation engine, each of the secondary threads running a corresponding interpreter that interprets an associated scripted routine, each scripted routine comprising an associated user-defined call that is mapped to a simulation engine task;

providing a plurality of user-defined simulation engine functions, each user-defined function associated with a corresponding interpreter;

upon encountering one of the user-defined functions, passing control from the simulation engine to the corresponding interpreter to interpret the associated scripted routine; and

upon encountering one of the user-defined calls, passing control from the interpreter to the simulation engine.